



Amazon
Linux

Servicios administrativo



INTRODUCCIÓN

La administración de servicios y el monitoreo en Linux son componentes vitales para mantener un sistema operativo estable, seguro y eficiente. Los servicios son programas que se ejecutan en segundo plano y proporcionan diversas funcionalidades esenciales, como servidores web, bases de datos, servicios de red y más. Administrar y monitorear estos servicios garantiza que estén funcionando correctamente y que cualquier problema se pueda detectar y resolver rápidamente.

OBJETIVOS

- Verificar el estado del servicio httpd para asegurarse de que se está ejecutando y que puede realizar una conexión http con la dirección IP del host local.
- También aprenderá a monitorear su instancia EC2 de Amazon Linux 2. Se utilizará el comando top de Linux y también el servicio AWS CloudWatch



TAREA 1

En esta tarea, se conectará a una instancia EC2 de Amazon Linux. Utilizará una utilidad SSH para realizar todas estas operaciones. Las siguientes instrucciones varían ligeramente según si utiliza Windows o Mac/Linux.

En Linux

- o Usando distribución Ubuntu con Subsistema de Windows para Linux (WSL).

```
ec2-user@ip-10-0-10-43:~ x + v
leps2408@LAPTOP-1I89QL1A:~$ neofetch
./+oossssoo+/- .
`:+ssssssssssssssssst+:`
  -+sssssssssssssssssyssst+-
  .ossssssssssssssssdMMMNysssso.
  /ssssssssssshdmmNNmyNMMMNhssssss/
  +ssssssssshmydMMMMMMNdddyssssssst+
  /ssssssssshNMMMyhhyyyyhNMMMNhssssss/
  .ssssssssdMMMNhssssssssshNMMMdssssss.
+sssshhhyNMMNysssssssssssyNMMMyssssst+
osssyNMMMNyMMhssssssssssshmmhssssssso
osssyNMMMNyMMhssssssssssshmmhssssssso
+sssshhhyNMMNysssssssssssyNMMMyssssst+
.sssssssdMMMNhssssssssshNMMMdssssss.
/ssssssssshNMMMyhhyyyyhNMMMNhssssss/
+ssssssssdmydMMMMMMNdddyssssssst+
/ssssssssshdmmNNNmyNMMMNhssssss/
  .ossssssssssssssssdMMMNysssso.
  -+sssssssssssssssssyssst+-
  `:+ssssssssssssssssst+:`
  ./+oossssoo+/- .

leps2408@LAPTOP-1I89QL1A
-----
OS: Ubuntu 20.04.6 LTS on Windows 10 x86_64
Kernel: 5.15.153.1-microsoft-standard-WSL2
Uptime: secs
Packages: 673 (dpkg), 4 (snap)
Shell: bash 5.0.17
Theme: Adwaita [GTK3]
Icons: Adwaita [GTK3]
Terminal: Relay(482)
CPU: Intel i5-10300H (8) @ 2.496GHz
GPU: 0929:00:00.0 Microsoft Corporation Device 008e
Memory: 421MiB / 3838MiB
```




TAREA 2

Httpd es el servicio para el servidor http Apache que está instalado en su host. Se trata de un servidor web ligero, como los que ejecutan sus sitios web favoritos (piense, por ejemplo, en amazon.com). En este ejercicio, se comprueba el estado del servicio httpd. Se debe iniciar utilizando el comando `systemctl` y se verifica que el servicio funciona.

- Compruebe el estado del servicio `httpd` utilizando los comandos `systemctl`, como se muestra a continuación. Nota: Servicio inactivo (dead).

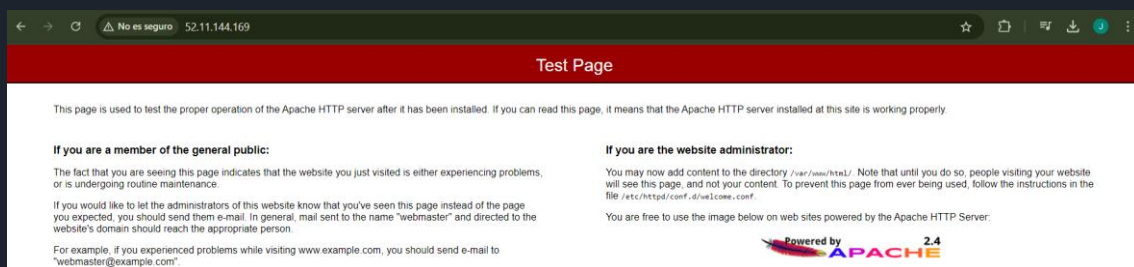
```
ec2-user@ip-10-0-10-81:~ x + v
[ec2-user@ip-10-0-10-81 ~]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[ec2-user@ip-10-0-10-81 ~]$ |
```

- Ingresar `sudo systemctl start httpd.service` para iniciar el servicio. Luego verificar el estado con el comando `sudo systemctl status httpd.service`.



```
ec2-user@ip-10-0-10-81:~  
[ec2-user@ip-10-0-10-81 ~]$ sudo systemctl start httpd.service  
[ec2-user@ip-10-0-10-81 ~]$ sudo systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)  
   Active: active (running) since Tue 2024-07-30 17:25:37 UTC; 16s ago  
     Docs: man:httpd.service(8)  
  Main PID: 2585 (httpd)  
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:  
0 B/sec"  
   CGroup: /system.slice/httpd.service  
           └─2585 /usr/sbin/httpd -DFOREGROUND  
             └─2586 /usr/sbin/httpd -DFOREGROUND  
               └─2588 /usr/sbin/httpd -DFOREGROUND  
                 └─2593 /usr/sbin/httpd -DFOREGROUND  
                   └─2595 /usr/sbin/httpd -DFOREGROUND  
                     └─2600 /usr/sbin/httpd -DFOREGROUND  
  
Jul 30 17:25:37 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Starting The Apache ...  
Jul 30 17:25:37 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Started The Apache H...  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-10-0-10-81 ~]$ |
```

- Abrir una nueva pestaña en el buscador en ingresar <http://<publicip>>, reemplazando con la IP pública.



- Ahora se puede para el servicio con `sudo systemctl stop httpd.service`.

```
ec2-user@ip-10-0-10-81:~  
[ec2-user@ip-10-0-10-81 ~]$ sudo systemctl stop httpd.service  
[ec2-user@ip-10-0-10-81 ~]$ sudo systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)  
   Active: inactive (dead)  
     Docs: man:httpd.service(8)  
  
Jul 30 17:25:37 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Starting The Apache ...  
Jul 30 17:25:37 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Started The Apache H...  
Jul 30 17:28:14 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Stopping The Apache ...  
Jul 30 17:28:15 ip-10-0-10-81.us-west-2.compute.internal systemd[1]: Stopped The Apache H...  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-10-0-10-81 ~]$ |
```



TAREA 3

En este ejercicio, se utilizará los comandos de Linux para monitorear la instancia EC2 de Amazon Linux2. También se abrirá la consola de AWS e iniciará sesión en CloudWatch para ver cómo este servicio puede proporcionarle datos para monitorear su instancia.

- o Ingresar el comando *top*. Top muestra los procesos que se están ejecutando en el momento y el uso de recursos, como el uso de la CPU y la memoria.

```
ec2-user@ip-10-0-10-81:~  
[ec2-user@ip-10-0-10-81 ~]$ top  
top - 17:29:23 up 9 min, 1 user, load average: 0.00, 0.02, 0.02  
Tasks: 87 total, 1 running, 47 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
KiB Mem : 966808 total, 438712 free, 76284 used, 451812 buff/cache  
KiB Swap: 0 total, 0 free, 0 used. 748196 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	123620	5592	3976	S	0.0	0.6	0:01.11	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	I	0.0	0.0	0:00.05	kworker/0:0
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.03	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.18	migration/1
16	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/1
17	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
22	root	20	0	0	0	0	I	0.0	0.0	0:00.16	kworker/u4:1
34	root	20	0	0	0	0	I	0.0	0.0	0:00.02	kworker/0:1
101	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u4:2

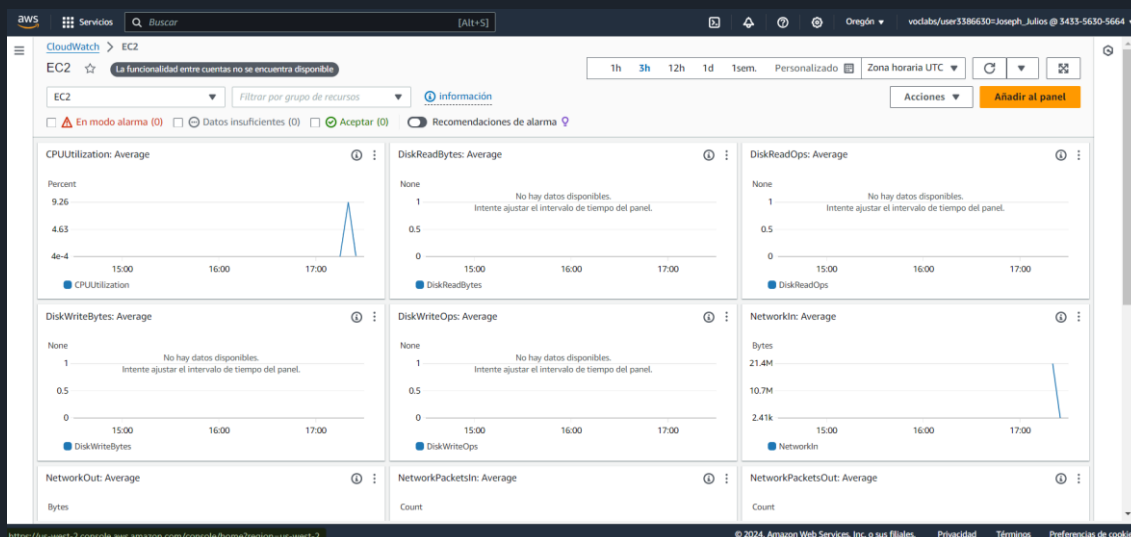


- Ejecutar el script stress.sh para simular una carga de trabajo pesada en la instancia de ec2. Ingresar también el comando top para mostrar los procesos.

```
ec2-user@ip-10-0-10-81:~$ ./stress.sh & top
[1] 2661
stress: info: [2663] dispatching hogs: 8 cpu, 4 io, 2 vm, 0 hdd
top - 17:30:45 up 11 min, 1 user, load average: 1.12, 0.25, 0.08
Tasks: 103 total, 15 running, 49 sleeping, 0 stopped, 0 zombie
%Cpu(s): 62.1 us, 37.9 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 966808 total, 369156 free, 145852 used, 451800 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 678636 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2672	ec2-user	20	0	7580	96	0	R	14.6	0.0	0:01.30	stress
2677	ec2-user	20	0	7580	96	0	R	14.6	0.0	0:01.30	stress
2665	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.29	stress
2666	ec2-user	20	0	138656	54420	208	R	14.3	5.6	0:01.29	stress
2667	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.29	stress
2668	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.29	stress
2669	ec2-user	20	0	138656	13236	208	R	14.3	1.4	0:01.29	stress
2670	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.30	stress
2671	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.29	stress
2673	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.29	stress
2674	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.30	stress
2676	ec2-user	20	0	7580	96	0	R	14.3	0.0	0:01.41	stress
2664	ec2-user	20	0	7580	96	0	R	14.0	0.0	0:01.29	stress
2675	ec2-user	20	0	7580	96	0	R	14.0	0.0	0:01.29	stress
1	root	20	0	123620	5592	3976	S	0.0	0.6	0:01.11	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	I	0.0	0.0	0:00.05	kworker/0:0
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/0

- Abrir AWS Management Console para utilizar Amazon CloudWatch. Seleccionar Automatic dashboards, en dashboard y elegir EC2.





- Revisar nuevamente el panel de CPUUtilization luego de 5 min. Nota: Debido al script de stress.sh, se ha alcanzado un porcentaje de utilización de 78.8% como máximo. Disminuye luego de 11 minutos.

